

APPENDIX A

```
%{
/*****
**
*   Copyright 1996, 1997, 1998 - A-LIFE Medical, Inc. - All Rights
Reserved.
*
*   Program: LifeCode Medical Coding System
*
*   File: temporal.lex
*
*   Description:
*       This file defines the parser for temporal expressions.
*
*   RCS Data: (do not edit)
*
*   Author:  $Author: mlm $
*   Date:    $Date: 1997/05/16 14:06:48 $
*   Version: $Revision: 1.2 $
*
*   History: $Log: temporal.lex $
*   Revision 1.2  1997/05/16  14:06:48  mlm
*   Added recognition of cardinal numbers 53-90
*
*****/

#include <iostream.h>
#include <string.h>

#ifdef LEXER
// TimeLexer class definition
extern int yyignore;

class TimeLexerClass {

private:

    char    begin_buf[256];
    char    end_buf[256];
    char    mod[256];
    char    modtext[256];
    char    end_int[256];
    ostream *ostr;

public:

    int temporal (char* scale, unsigned char* text) {
        if (yyignore) {
            not_temporal(text);
            return 0;
        }
    }
};
```



```

    }
    *ostr << " ---T " << mod << " " << end_int << " " << scale << "
T--- ";
    begin_buf[0] = NULL;
    end_buf[0] = NULL;
    mod[0] = NULL;
    modtext[0] = NULL;
    return 1;
}

int number (char *week, unsigned char* text) {
    if (yyignore) {
        not_temporal(text);
        return 0;
    }
    if (begin_buf[0] == NULL) {
        strcpy(begin_buf, (char *)text);
        strcpy(end_int, week);
    } else {
        strcpy(end_buf, (char *)text);
        strcpy(end_int, week);
        if (!strcmp(begin_buf, end_buf)) begin_buf[0] = NULL;
    }
    return 1;
}

int push (char* scale, unsigned char* text) {
    if (yyignore) {
        not_temporal(text);
        return 0;
    }
    strcpy(modtext, (char *)text);
    if (strcmp(scale, "JUNK")) {
        strcpy(mod, scale);
    }
    return 1;
}

int not_temporal (unsigned char* text) {
    *ostr << modtext << begin_buf << end_buf << (char *)text;
    clear();
    return 1;
}

int clear (void) {
    begin_buf[0] = NULL;
    end_buf[0] = NULL;
    mod[0] = NULL;
    modtext[0] = NULL;
    end_int[0] = NULL;
    return 1;
}

void end_line(void) { *ostr << endl; }

TimeLexerClass() {
    ostr = &cout;
}

```

```

        begin_buf[0] = NULL;
        end_buf[0] = NULL;
        mod[0] = NULL;
        modtext[0] = NULL;
        end_int[0] = NULL;
    }

    ~TimeLexerClass() {
    }
};

// Segmenter Base class definition
class SegmenterBaseClass {

public:

    TimeLexerClass time_lexer;

    SegmenterBaseClass() {
    }

    ~SegmenterBaseClass() {
    }
};

#endif

#ifndef LEXER
#include "seg_base.h"
#undef YY_INPUT
#define YY_INPUT(buf,result,max_size) \
    Segmenter_p->time_lexer.get(buf, max_size, EOF); \
    result = Segmenter_p->time_lexer.gcount(); \
    Segmenter_p->time_lexer.accumulate(result);
#endif

extern SegmenterBaseClass *Segmenter_p;
extern int yylex();
int yyignore = 0;

%}

LT          (less[ ]than|within[ ](the[ ]last)?|not?[ ]more[
]than)
MT          ((more|longer)[ ]than|over|at[ ]?least|not?[ ]less[
]than)
JUNK        (approximately|about|around)
AGO         (ago|since|past)
MN1         (minute([ ]|s[ ]|s' [ ]|'s[ ]){AGO}?)
MN          ([\ - ]?{MN1})
HR1         (hour([ ]|s[ ]|s' [ ]|'s[ ]){AGO}?)
HR          ([\ - ]?{HR1})
DA1         (day([ ]|s[ ]|s' [ ]|'s[ ]){AGO}?)
DA          ([\ - ]?{DA1})
WK1         (we?e?k([ ]|s[ ]|s' [ ]|'s[ ]){AGO}?)
WK          ([\ - ]?{WK1})
MO1         (month([ ]|s[ ]|s' [ ]|'s[ ]){AGO}?)
MO          ([\ - ]?{MO1})

```

YR1	(year([] s[] s'[] s'[]){AGO}?)
YR	([\-]?{YR1})
DASH	([]?-[]? [](to or)[])
ND	({DASH}?)
DASH_SP	({ND}({DASH} []))
HALF	([\-]?1\2 ((one a)[\-])?half)
QUARTER	([\-]?1\4 ((one a)[\-])?quarter)
HALF1	([\-]?1\2 (one a)[\-]half)
QUARTER1	([\-]?1\4 (one a)[\-]quarter)
FRACTION	({QUARTER1} {HALF1})
FRACT	([]?(and)?[]?{FRACTION})
FRACT_HALF	([]?(and)?[]?{HALF1})
FRACT_QUARTER	([]?(and)?[]?{QUARTER1})

_0_1	({DASH_SP}0?\1[0-9]*)
_0_2	({DASH_SP}0?\2[0-4]*)
_0_25	({DASH_SP}(0?\2[5-9] {QUARTER}))
_0_3	({DASH_SP}0?\3[0-9]*)
_0_4	({DASH_SP}0?\4[0-9]*)
_0_5	({DASH_SP}(0?\5[0-9]* {HALF}))
_0_6	({DASH_SP}0?\6[0-9]*)
_0_7	({DASH_SP}0?\7[0-4]*)
_0_75	({DASH_SP}0?\7[5-9])
_0_8	({DASH_SP}0?\8[0-9]*)
_0_9	({DASH_SP}0?\9[0-9]*)
_1	({DASH_SP}(1 one an a))
_1_1	({DASH_SP}1\1[0-9]*)
_1_2	({DASH_SP}1\2[0-4]*)
_1_25	({DASH_SP}(1\2[5-9] (1 one){FRACT_QUARTER}))
_1_25HR	({DASH_SP}(1 one an)[\-]hours?{FRACT_QUARTER})
_1_25DA	({DASH_SP}(1 one)[\-]days?{FRACT_QUARTER})
_1_25WK	({DASH_SP}(1 one)[\-]we?e?ks?{FRACT_QUARTER})
_1_3	({DASH_SP}1\3[0-9]*)
_1_4	({DASH_SP}1\4[0-9]*)
_1_5	({DASH_SP}(1\5[0-9]* (1 one){FRACT_HALF}))
_1_5HR	({DASH_SP}(1 one an)[\-]hours?{FRACT_HALF})
_1_5DA	({DASH_SP}(1 one)[\-]days?{FRACT_HALF})
_1_5WK	({DASH_SP}(1 one)[\-]we?e?ks?{FRACT_HALF})
_1_6	({DASH_SP}1\6[0-9]*)
_1_7	({DASH_SP}1\7[0-4]*)
_1_75	({DASH_SP}1\7[5-9])
_1_8	({DASH_SP}1\8[0-9]*)
_1_9	({DASH_SP}1\9[0-9]*)
_2	({DASH_SP}(2 two))
_2_1	({DASH_SP}2\1[0-9]*)
_2_2	({DASH_SP}2\2[0-4]*)
_2_25	({DASH_SP}(2\2[5-9] (2 two){FRACT_QUARTER}))
_2_25HR	({DASH_SP}(2 two)[\-]hours?{FRACT_QUARTER})
_2_25DA	({DASH_SP}(2 two)[\-]days?{FRACT_QUARTER})
_2_25WK	({DASH_SP}(2 two)[\-]we?e?ks?{FRACT_QUARTER})
_2_3	({DASH_SP}2\3[0-9]*)
_2_4	({DASH_SP}2\4[0-9]*)
_2_5	({DASH_SP}(2\5[0-9]* (2 two){FRACT_HALF}))
_2_5HR	({DASH_SP}(2 two)[\-]hours?{FRACT_HALF})
_2_5DA	({DASH_SP}(2 two)[\-]days?{FRACT_HALF})
_2_5WK	({DASH_SP}(2 two)[\-]we?e?ks?{FRACT_HALF})
_2_6	({DASH_SP}2\6[0-9]*)

_2_7	({DASH_SP} 2 \ . 7 [0 - 4] *)
_2_75	({DASH_SP} 2 \ . 7 [5 - 9])
_2_8	({DASH_SP} 2 \ . 8 [0 - 9] *)
_2_9	({DASH_SP} 2 \ . 9 [0 - 9] *)
_3	({DASH_SP} (3 three))
_3_1	({DASH_SP} 3 \ . 1 [0 - 9] *)
_3_2	({DASH_SP} 3 \ . 2 [0 - 4] *)
_3_25	({DASH_SP} (3 \ . 2 [5 - 9] (3 three) { FRACT_QUARTER }))
_3_25HR	({DASH_SP} (3 three) [\ -] hours ? { FRACT_QUARTER })
_3_25DA	({DASH_SP} (3 three) [\ -] days ? { FRACT_QUARTER })
_3_25WK	({DASH_SP} (3 three) [\ -] we ? e ? ks ? { FRACT_QUARTER })
_3_3	({DASH_SP} 3 \ . 3 [0 - 9] *)
_3_4	({DASH_SP} 3 \ . 4 [0 - 9] *)
_3_5	({DASH_SP} (3 \ . 5 [0 - 9] * (3 three) { FRACT_HALF }))
_3_5HR	({DASH_SP} (3 three) [\ -] hours ? { FRACT_HALF })
_3_5DA	({DASH_SP} (3 three) [\ -] days ? { FRACT_HALF })
_3_5WK	({DASH_SP} (3 three) [\ -] we ? e ? ks ? { FRACT_HALF })
_3_6	({DASH_SP} 3 \ . 6 [0 - 9] *)
_3_7	({DASH_SP} 3 \ . 7 [0 - 4] *)
_3_75	({DASH_SP} 3 \ . 7 [5 - 9])
_3_8	({DASH_SP} 3 \ . 8 [0 - 9] *)
_3_9	({DASH_SP} 3 \ . 9 [0 - 9] *)
_4	({DASH_SP} (4 four))
_4_1	({DASH_SP} 4 \ . 1 [0 - 9] *)
_4_2	({DASH_SP} 4 \ . 2 [0 - 4] *)
_4_25	({DASH_SP} (4 \ . 2 [5 - 9] (4 four) { FRACT_QUARTER }))
_4_25HR	({DASH_SP} (4 four) [\ -] hours ? { FRACT_QUARTER })
_4_25DA	({DASH_SP} (4 four) [\ -] days ? { FRACT_QUARTER })
_4_25WK	({DASH_SP} (4 four) [\ -] we ? e ? ks ? { FRACT_QUARTER })
_4_3	({DASH_SP} 4 \ . 3 [0 - 9] *)
_4_4	({DASH_SP} 4 \ . 4 [0 - 9] *)
_4_5	({DASH_SP} (4 \ . 5 [0 - 9] * (4 four) { FRACT_HALF }))
_4_5HR	({DASH_SP} (4 four) [\ -] hours ? { FRACT_HALF })
_4_5DA	({DASH_SP} (4 four) [\ -] days ? { FRACT_HALF })
_4_5WK	({DASH_SP} (4 four) [\ -] we ? e ? ks ? { FRACT_HALF })
_4_6	({DASH_SP} 4 \ . 6 [0 - 9] *)
_4_7	({DASH_SP} 4 \ . 7 [0 - 4] *)
_4_75	({DASH_SP} 4 \ . 7 [5 - 9])
_4_8	({DASH_SP} 4 \ . 8 [0 - 9] *)
_4_9	({DASH_SP} 4 \ . 9 [0 - 9] *)
_5	({DASH_SP} (5 five))
_5_1	({DASH_SP} 5 \ . 1 [0 - 9] *)
_5_2	({DASH_SP} 5 \ . 2 [0 - 4] *)
_5_25	({DASH_SP} (5 \ . 2 [5 - 9] (5 five) { FRACT_QUARTER }))
_5_25HR	({DASH_SP} (5 five) [\ -] hours ? { FRACT_QUARTER })
_5_25DA	({DASH_SP} (5 five) [\ -] days ? { FRACT_QUARTER })
_5_25WK	({DASH_SP} (5 five) [\ -] we ? e ? ks ? { FRACT_QUARTER })
_5_3	({DASH_SP} 5 \ . 3 [0 - 9] *)
_5_4	({DASH_SP} 5 \ . 4 [0 - 9] *)
_5_5	({DASH_SP} (5 \ . 5 [0 - 9] * (5 five) { FRACT_HALF }))
_5_5HR	({DASH_SP} (5 five) [\ -] hours ? { FRACT_HALF })
_5_5DA	({DASH_SP} (5 five) [\ -] days ? { FRACT_HALF })
_5_5WK	({DASH_SP} (5 five) [\ -] we ? e ? ks ? { FRACT_HALF })
_5_6	({DASH_SP} 5 \ . 6 [0 - 9] *)
_5_7	({DASH_SP} 5 \ . 7 [0 - 4] *)
_5_75	({DASH_SP} 5 \ . 7 [5 - 9])
_5_8	({DASH_SP} 5 \ . 8 [0 - 9] *)

_5_9	{{DASH_SP}5\ .9[0-9]*}
_6	{{DASH_SP}(6 six)}
_6_1	{{DASH_SP}6\ .1[0-9]*}
_6_2	{{DASH_SP}6\ .2[0-4]*}
_6_25	{{DASH_SP}(6\ .2[5-9] (6 six){FRACT_QUARTER}})
_6_25HR	{{DASH_SP}(6 six)[\ -]hours?{FRACT_QUARTER}}
_6_25DA	{{DASH_SP}(6 six)[\ -]days?{FRACT_QUARTER}}
_6_25WK	{{DASH_SP}(6 six)[\ -]we?e?ks?{FRACT_QUARTER}}
_6_3	{{DASH_SP}6\ .3[0-9]*}
_6_4	{{DASH_SP}6\ .4[0-9]*}
_6_5	{{DASH_SP}(6\ .5[0-9]* (6 six){FRACT_HALF}})
_6_5HR	{{DASH_SP}(6 six)[\ -]hours?{FRACT_HALF}}
_6_5DA	{{DASH_SP}(6 six)[\ -]days?{FRACT_HALF}}
_6_5WK	{{DASH_SP}(6 six)[\ -]we?e?ks?{FRACT_HALF}}
_6_6	{{DASH_SP}6\ .6[0-9]*}
_6_7	{{DASH_SP}6\ .7[0-4]*}
_6_75	{{DASH_SP}6\ .7[5-9]}
_6_8	{{DASH_SP}6\ .8[0-9]*}
_6_9	{{DASH_SP}6\ .9[0-9]*}
_7	{{DASH_SP}(7 seven)}
_7_1	{{DASH_SP}7\ .1[0-9]*}
_7_2	{{DASH_SP}7\ .2[0-4]*}
_7_25	{{DASH_SP}(7\ .2[5-9] (7 seven){FRACT_QUARTER}})
_7_25HR	{{DASH_SP}(7 seven)[\ -]hours?{FRACT_QUARTER}}
_7_25DA	{{DASH_SP}(7 seven)[\ -]days?{FRACT_QUARTER}}
_7_25WK	{{DASH_SP}(7 seven)[\ -]we?e?ks?{FRACT_QUARTER}}
_7_3	{{DASH_SP}7\ .3[0-9]*}
_7_4	{{DASH_SP}7\ .4[0-9]*}
_7_5	{{DASH_SP}(7\ .5[0-9]* (7 seven){FRACT_HALF}})
_7_5HR	{{DASH_SP}(7 seven)[\ -]hours?{FRACT_HALF}}
_7_5DA	{{DASH_SP}(7 seven)[\ -]days?{FRACT_HALF}}
_7_5WK	{{DASH_SP}(7 seven)[\ -]we?e?ks?{FRACT_HALF}}
_7_6	{{DASH_SP}7\ .6[0-9]*}
_7_7	{{DASH_SP}7\ .7[0-4]*}
_7_75	{{DASH_SP}7\ .7[5-9]}
_7_8	{{DASH_SP}7\ .8[0-9]*}
_7_9	{{DASH_SP}7\ .9[0-9]*}
_8	{{DASH_SP}(8 eight)}
_8_1	{{DASH_SP}8\ .1[0-9]*}
_8_2	{{DASH_SP}8\ .2[0-4]*}
_8_25	{{DASH_SP}(8\ .2[5-9] (8 eight){FRACT_QUARTER}})
_8_25HR	{{DASH_SP}(8 eight)[\ -]hours?{FRACT_QUARTER}}
_8_25DA	{{DASH_SP}(8 eight)[\ -]days?{FRACT_QUARTER}}
_8_25WK	{{DASH_SP}(8 eight)[\ -]we?e?ks?{FRACT_QUARTER}}
_8_3	{{DASH_SP}8\ .3[0-9]*}
_8_4	{{DASH_SP}8\ .4[0-9]*}
_8_5	{{DASH_SP}(8\ .5[0-9]* (8 eight){FRACT_HALF}})
_8_5HR	{{DASH_SP}(8 eight)[\ -]hours?{FRACT_HALF}}
_8_5DA	{{DASH_SP}(8 eight)[\ -]days?{FRACT_HALF}}
_8_5WK	{{DASH_SP}(8 eight)[\ -]we?e?ks?{FRACT_HALF}}
_8_6	{{DASH_SP}8\ .6[0-9]*}
_8_7	{{DASH_SP}8\ .7[0-4]*}
_8_75	{{DASH_SP}8\ .7[5-9]}
_8_8	{{DASH_SP}8\ .8[0-9]*}
_8_9	{{DASH_SP}8\ .9[0-9]*}
_9	{{DASH_SP}(9 nine)}
_9_1	{{DASH_SP}9\ .1[0-9]*}

_9_2 ({DASH_SP}9\.2[0-4]*)
 _9_25 ({DASH_SP}(9\.2[5-9]|(9|nine){FRACT_QUARTER}))
 _9_25HR ({DASH_SP}(9|nine)[\ -]hours?{FRACT_QUARTER})
 _9_25DA ({DASH_SP}(9|nine)[\ -]days?{FRACT_QUARTER})
 _9_25WK ({DASH_SP}(9|nine)[\ -]we?e?ks?{FRACT_QUARTER})
 _9_3 ({DASH_SP}9\.3[0-9]*)
 _9_4 ({DASH_SP}9\.4[0-9]*)
 _9_5 ({DASH_SP}(9\.5[0-9]*|(9|nine){FRACT_HALF}))
 _9_5HR ({DASH_SP}(9|nine)[\ -]hours?{FRACT_HALF})
 _9_5DA ({DASH_SP}(9|nine)[\ -]days?{FRACT_HALF})
 _9_5WK ({DASH_SP}(9|nine)[\ -]we?e?ks?{FRACT_HALF})
 _9_6 ({DASH_SP}9\.6[0-9]*)
 _9_7 ({DASH_SP}9\.7[0-4]*)
 _9_75 ({DASH_SP}9\.7[5-9])
 _9_8 ({DASH_SP}9\.8[0-9]*)
 _9_9 ({DASH_SP}9\.9[0-9]*)
 _10 ({DASH_SP}(10|ten))
 _11 ({DASH_SP}(11|10\. [0-9]+|eleven|(10|ten){FRACT}))
 _12 ({DASH_SP}(12|11\. [0-9]+|twelve|(11|eleven){FRACT}))
 _13 ({DASH_SP}(13|12\. [0-9]+|thirteen|(12|twelve){FRACT}))
 _14 ({DASH_SP}(14|13\. [0-9]+|fourteen|(13|thirteen){FRACT}))
 _15 ({DASH_SP}(15|14\. [0-9]+|fifteen|(14|fourteen){FRACT}))
 _16 ({DASH_SP}(16|15\. [0-9]+|sixteen|(15|fifteen){FRACT}))
 _17 ({DASH_SP}(17|16\. [0-9]+|seventeen|(16|sixteen){FRACT}))
 _18 ({DASH_SP}(18|17\. [0-9]+|eighteen|(17|seventeen){FRACT}))
 _19 ({DASH_SP}(19|18\. [0-9]+|nineteen|(18|eighteen){FRACT}))
 _20 ({DASH_SP}(20|19\. [0-9]+|twenty|(19|nineteen){FRACT}))
 _21 ({DASH_SP}(21|20\. [0-9]+|twenty-one|(20|twenty){FRACT}))
 _22 ({DASH_SP}(22|21\. [0-9]+|twenty-two|(21|twenty-one){FRACT}))
 _23 ({DASH_SP}(23|22\. [0-9]+|twenty-three|(22|twenty-two){FRACT}))
 _24 ({DASH_SP}(24|23\. [0-9]+|twenty-four|(23|twenty-three){FRACT}))
 _25 ({DASH_SP}(25|24\. [0-9]+|twenty-five|(24|twenty-four){FRACT}))
 _26 ({DASH_SP}(26|25\. [0-9]+|twenty-six|(25|twenty-five){FRACT}))
 _27 ({DASH_SP}(27|26\. [0-9]+|twenty-seven|(26|twenty-six){FRACT}))
 _28 ({DASH_SP}(28|27\. [0-9]+|twenty-eight|(27|twenty-seven){FRACT}))
 _29 ({DASH_SP}(29|28\. [0-9]+|twenty-nine|(28|twenty-eight){FRACT}))
 _30 ({DASH_SP}(30|29\. [0-9]+|thirty|(29|twenty-nine){FRACT}))
 _31 ({DASH_SP}(31|30\. [0-9]+|thirty-one|(30|thirty){FRACT}))

_32	((DASH_SP}{32 31\[. [0-9]+ thirty-?two (31 thirty-
?one){FRACT}))	
_33	((DASH_SP}{33 32\[. [0-9]+ thirty-?three (32 thirty-
?two){FRACT}))	
_34	((DASH_SP}{34 33\[. [0-9]+ thirty-?four (33 thirty-
?three){FRACT}))	
_35	((DASH_SP}{35 34\[. [0-9]+ thirty-?five (34 thirty-
?four){FRACT}))	
_36	((DASH_SP}{36 35\[. [0-9]+ thirty-?six (35 thirty-
?five){FRACT}))	
_37	((DASH_SP}{37 36\[. [0-9]+ thirty-?seven (36 thirty-
?six){FRACT}))	
_38	((DASH_SP}{38 37\[. [0-9]+ thirty-?eight (37 thirty-
?seven){FRACT}))	
_39	((DASH_SP}{39 38\[. [0-9]+ thirty-?nine (38 thirty-
?eight){FRACT}))	
_40	((DASH_SP}{40 39\[. [0-9]+ fou?rty (39 thirty-
?nine){FRACT}))	
_41	((DASH_SP}{41 40\[. [0-9]+ fou?rty-
?one (40 fou?rty){FRACT}))	
_42	((DASH_SP}{42 41\[. [0-9]+ fou?rty-?two (41 fou?rty-
?one){FRACT}))	
_43	((DASH_SP}{43 42\[. [0-9]+ fou?rty-?three (42 fou?rty-
?two){FRACT}))	
_44	((DASH_SP}{44 43\[. [0-9]+ fou?rty-?four (43 fou?rty-
?three){FRACT}))	
_45	((DASH_SP}{45 44\[. [0-9]+ fou?rty-?five (44 fou?rty-
?four){FRACT}))	
_46	((DASH_SP}{46 45\[. [0-9]+ fou?rty-?six (45 fou?rty-
?five){FRACT}))	
_47	((DASH_SP}{47 46\[. [0-9]+ fou?rty-?seven (46 fou?rty-
?six){FRACT}))	
_48	((DASH_SP}{48 47\[. [0-9]+ fou?rty-?eight (47 fou?rty-
?seven){FRACT}))	
_49	((DASH_SP}{49 48\[. [0-9]+ fou?rty-?nine (48 fou?rty-
?eight){FRACT}))	
_50	((DASH_SP}{50 49\[. [0-9]+ fifty (49 fou?rty-
?nine){FRACT}))	
_51	((DASH_SP}{51 50\[. [0-9]+ fifty-
?one (50 fifty){FRACT}))	
_52	((DASH_SP}{52 51\[. [0-9]+ fifty-?two (51 fifty-
?one){FRACT}))	
_53	((DASH_SP}{53 52\[. [0-9]+ fifty-?three (52 fifty-
?two){FRACT}))	
_54	((DASH_SP}{54 53\[. [0-9]+ fifty-?four (53 fifty-
?three){FRACT}))	
_55	((DASH_SP}{55 54\[. [0-9]+ fifty-?five (54 fifty-
?four){FRACT}))	
_56	((DASH_SP}{56 55\[. [0-9]+ fifty-?six (55 fifty-
?five){FRACT}))	
_57	((DASH_SP}{57 56\[. [0-9]+ fifty-?seven (56 fifty-
?six){FRACT}))	
_58	((DASH_SP}{58 57\[. [0-9]+ fifty-?eight (57 fifty-
?seven){FRACT}))	
_59	((DASH_SP}{59 58\[. [0-9]+ fifty-?nine (58 fifty-
?eight){FRACT}))	

_60 ({DASH_SP} (60|59\.[0-9]+|sixty| (59|fifty-
 ?nine) {FRACT}))
 _61 ({DASH_SP} (61|60\.[0-9]+|sixty-
 ?one| (60|sixty) {FRACT}))
 _62 ({DASH_SP} (62|61\.[0-9]+|sixty-?two| (61|sixty-
 ?one) {FRACT}))
 _63 ({DASH_SP} (63|62\.[0-9]+|sixty-?three| (62|sixty-
 ?two) {FRACT}))
 _64 ({DASH_SP} (64|63\.[0-9]+|sixty-?four| (63|sixty-
 ?three) {FRACT}))
 _65 ({DASH_SP} (65|64\.[0-9]+|sixty-?five| (64|sixty-
 ?four) {FRACT}))
 _66 ({DASH_SP} (66|65\.[0-9]+|sixty-?six| (65|sixty-
 ?five) {FRACT}))
 _67 ({DASH_SP} (67|66\.[0-9]+|sixty-?seven| (66|sixty-
 ?six) {FRACT}))
 _68 ({DASH_SP} (68|67\.[0-9]+|sixty-?eight| (67|sixty-
 ?seven) {FRACT}))
 _69 ({DASH_SP} (69|68\.[0-9]+|sixty-?nine| (68|sixty-
 ?eight) {FRACT}))
 _70 ({DASH_SP} (70|69\.[0-9]+|seventy| (69|sixty-
 ?nine) {FRACT}))
 _71 ({DASH_SP} (71|70\.[0-9]+|seventy-
 ?one| (70|seventy) {FRACT}))
 _72 ({DASH_SP} (72|71\.[0-9]+|seventy-?two| (71|seventy-
 ?one) {FRACT}))
 _73 ({DASH_SP} (73|72\.[0-9]+|seventy-?three| (72|seventy-
 ?two) {FRACT}))
 _74 ({DASH_SP} (74|73\.[0-9]+|seventy-?four| (73|seventy-
 ?three) {FRACT}))
 _75 ({DASH_SP} (75|74\.[0-9]+|seventy-?five| (74|seventy-
 ?four) {FRACT}))
 _76 ({DASH_SP} (76|75\.[0-9]+|seventy-?six| (75|seventy-
 ?five) {FRACT}))
 _77 ({DASH_SP} (77|76\.[0-9]+|seventy-?seven| (76|seventy-
 ?six) {FRACT}))
 _78 ({DASH_SP} (78|77\.[0-9]+|seventy-?eight| (77|seventy-
 ?seven) {FRACT}))
 _79 ({DASH_SP} (79|78\.[0-9]+|seventy-?nine| (78|seventy-
 ?eight) {FRACT}))
 _80 ({DASH_SP} (80|79\.[0-9]+|eighty| (79|seventy-
 ?nine) {FRACT}))
 _81 ({DASH_SP} (81|80\.[0-9]+|eighty-
 ?one| (80|eighty) {FRACT}))
 _82 ({DASH_SP} (82|81\.[0-9]+|eighty-?two| (81|eighty-
 ?one) {FRACT}))
 _83 ({DASH_SP} (83|82\.[0-9]+|eighty-?three| (82|eighty-
 ?two) {FRACT}))
 _84 ({DASH_SP} (84|83\.[0-9]+|eighty-?four| (83|eighty-
 ?three) {FRACT}))
 _85 ({DASH_SP} (85|84\.[0-9]+|eighty-?five| (84|eighty-
 ?four) {FRACT}))
 _86 ({DASH_SP} (86|85\.[0-9]+|eighty-?six| (85|eighty-
 ?five) {FRACT}))
 _87 ({DASH_SP} (87|86\.[0-9]+|eighty-?seven| (86|eighty-
 ?six) {FRACT}))

```

_88      ({DASH_SP} (88|87\.[0-9]+|eighty-?eight| (87|eighty-
?seven) {FRACT})))
_89      ({DASH_SP} (89|88\.[0-9]+|eighty-?nine| (88|eighty-
?eight) {FRACT})))
_90      ({DASH_SP} (90|89\.[0-9]+|ninety| (89|eighty-
?nine) {FRACT})))
_91      ({DASH_SP} 91)
_92      ({DASH_SP} 92)
_93      ({DASH_SP} 93)
_94      ({DASH_SP} 94)
_95      ({DASH_SP} 95)
_96      ({DASH_SP} 96)
_97      ({DASH_SP} 97)
_98      ({DASH_SP} 98)
_99      ({DASH_SP} 99)
_100s    ({DASH_SP} 1[0-9][0-9])

_1s      ({_1}|{_2}|{_3}|{_4}|{_5}|{_6}|{_7}|{_8}|{_9})
_10s     ({_10}|{_11}|{_12}|{_13}|{_14}|{_15}|{_16}|{_17}|{_18}|{_19}
})
_20s     ({_20}|{_21}|{_22}|{_23}|{_24}|{_25}|{_26}|{_27}|{_28}|{_29}
})
_30s     ({_30}|{_31}|{_32}|{_33}|{_34}|{_35}|{_36}|{_37}|{_38}|{_39}
})
_40s     ({_40}|{_41}|{_42}|{_43}|{_44}|{_45}|{_46}|{_47}|{_48}|{_49}
})
_50s     ({_50}|{_51}|{_52}|{_53}|{_54}|{_55}|{_56}|{_57}|{_58}|{_59}
})
_60s     ({_60}|{_61}|{_62}|{_63}|{_64}|{_65}|{_66}|{_67}|{_68}|{_69}
})
_70s     ({_70}|{_71}|{_72}|{_73}|{_74}|{_75}|{_76}|{_77}|{_78}|{_79}
})
_80s     ({_80}|{_81}|{_82}|{_83}|{_84}|{_85}|{_86}|{_87}|{_88}|{_89}
})
_90s     ({_90}|{_91}|{_92}|{_93}|{_94}|{_95}|{_96}|{_97}|{_98}|{_99}
})
NUM
({_1s}|{_10s}|{_20s}|{_30s}|{_40s}|{_50s}|{_60s}|{_70s}|{_80s}|{_90s}|{
_100s})

```

%%

```

---S      { yyignore = 1; Segmenter_p-
>time_lexer.not_temporal(yytext); }
S---      { yyignore = 0; Segmenter_p-
>time_lexer.not_temporal(yytext); }
---L      { yyignore = 1; Segmenter_p-
>time_lexer.not_temporal(yytext); }

```

```

L---      { yyignore = 0; Segmenter_p-
>time_lexer.not_temporal(yytext); }
{MN}      { if (yytext[yyleng-1] == ' ') yyless(yyleng-1);
            Segmenter_p->time_lexer.temporal("MN", yytext); }
{HR}      { if (yytext[yyleng-1] == ' ') yyless(yyleng-1);
            Segmenter_p->time_lexer.temporal("HR", yytext); }
{DA}      { if (yytext[yyleng-1] == ' ') yyless(yyleng-1);
            Segmenter_p->time_lexer.temporal("DA", yytext); }
{WK}      { if (yytext[yyleng-1] == ' ') yyless(yyleng-1);
            Segmenter_p->time_lexer.temporal("WK", yytext); }
{MO}      { if (yytext[yyleng-1] == ' ') yyless(yyleng-1);
            Segmenter_p->time_lexer.temporal("MO", yytext); }
{YR}      { if (yytext[yyleng-1] == ' ') yyless(yyleng-1);
            Segmenter_p->time_lexer.temporal("YR", yytext); }
{LT}      { Segmenter_p->time_lexer.push("LT", yytext); }
{MT}      { Segmenter_p->time_lexer.push("MT", yytext); }
{JUNK}    { Segmenter_p->time_lexer.push("JUNK", yytext); }
{_100s}   { Segmenter_p->time_lexer.number((char *)yytext, yytext); }
{_99}     { Segmenter_p->time_lexer.number("99", yytext); }
{_98}     { Segmenter_p->time_lexer.number("98", yytext); }
{_97}     { Segmenter_p->time_lexer.number("97", yytext); }
{_96}     { Segmenter_p->time_lexer.number("96", yytext); }
{_95}     { Segmenter_p->time_lexer.number("95", yytext); }
{_94}     { Segmenter_p->time_lexer.number("94", yytext); }
{_93}     { Segmenter_p->time_lexer.number("93", yytext); }
{_92}     { Segmenter_p->time_lexer.number("92", yytext); }
{_91}     { Segmenter_p->time_lexer.number("91", yytext); }
{_90}     { Segmenter_p->time_lexer.number("90", yytext); }
{_89}     { Segmenter_p->time_lexer.number("89", yytext); }
{_88}     { Segmenter_p->time_lexer.number("88", yytext); }
{_87}     { Segmenter_p->time_lexer.number("87", yytext); }
{_86}     { Segmenter_p->time_lexer.number("86", yytext); }
{_85}     { Segmenter_p->time_lexer.number("85", yytext); }
{_84}     { Segmenter_p->time_lexer.number("84", yytext); }
{_83}     { Segmenter_p->time_lexer.number("83", yytext); }
{_82}     { Segmenter_p->time_lexer.number("82", yytext); }
{_81}     { Segmenter_p->time_lexer.number("81", yytext); }
{_80}     { Segmenter_p->time_lexer.number("80", yytext); }
{_79}     { Segmenter_p->time_lexer.number("79", yytext); }
{_78}     { Segmenter_p->time_lexer.number("78", yytext); }
{_77}     { Segmenter_p->time_lexer.number("77", yytext); }
{_76}     { Segmenter_p->time_lexer.number("76", yytext); }
{_75}     { Segmenter_p->time_lexer.number("75", yytext); }
{_74}     { Segmenter_p->time_lexer.number("74", yytext); }
{_73}     { Segmenter_p->time_lexer.number("73", yytext); }
{_72}     { Segmenter_p->time_lexer.number("72", yytext); }
{_71}     { Segmenter_p->time_lexer.number("71", yytext); }
{_70}     { Segmenter_p->time_lexer.number("70", yytext); }
{_69}     { Segmenter_p->time_lexer.number("69", yytext); }
{_68}     { Segmenter_p->time_lexer.number("68", yytext); }
{_67}     { Segmenter_p->time_lexer.number("67", yytext); }
{_66}     { Segmenter_p->time_lexer.number("66", yytext); }
{_65}     { Segmenter_p->time_lexer.number("65", yytext); }
{_64}     { Segmenter_p->time_lexer.number("64", yytext); }
{_63}     { Segmenter_p->time_lexer.number("63", yytext); }
{_62}     { Segmenter_p->time_lexer.number("62", yytext); }
{_61}     { Segmenter_p->time_lexer.number("61", yytext); }

```

```

{ _60 } { Segmenter_p->time_lexer.number("60", yytext); }
{ _59 } { Segmenter_p->time_lexer.number("59", yytext); }
{ _58 } { Segmenter_p->time_lexer.number("58", yytext); }
{ _57 } { Segmenter_p->time_lexer.number("57", yytext); }
{ _56 } { Segmenter_p->time_lexer.number("56", yytext); }
{ _55 } { Segmenter_p->time_lexer.number("55", yytext); }
{ _54 } { Segmenter_p->time_lexer.number("54", yytext); }
{ _53 } { Segmenter_p->time_lexer.number("53", yytext); }
{ _52 } { Segmenter_p->time_lexer.number("52", yytext); }
{ _51 } { Segmenter_p->time_lexer.number("51", yytext); }
{ _50 } { Segmenter_p->time_lexer.number("50", yytext); }
{ _49 } { Segmenter_p->time_lexer.number("49", yytext); }
{ _48 } { Segmenter_p->time_lexer.number("48", yytext); }
{ _47 } { Segmenter_p->time_lexer.number("47", yytext); }
{ _46 } { Segmenter_p->time_lexer.number("46", yytext); }
{ _45 } { Segmenter_p->time_lexer.number("45", yytext); }
{ _44 } { Segmenter_p->time_lexer.number("44", yytext); }
{ _43 } { Segmenter_p->time_lexer.number("43", yytext); }
{ _42 } { Segmenter_p->time_lexer.number("42", yytext); }
{ _41 } { Segmenter_p->time_lexer.number("41", yytext); }
{ _40 } { Segmenter_p->time_lexer.number("40", yytext); }
{ _39 } { Segmenter_p->time_lexer.number("39", yytext); }
{ _38 } { Segmenter_p->time_lexer.number("38", yytext); }
{ _37 } { Segmenter_p->time_lexer.number("37", yytext); }
{ _36 } { Segmenter_p->time_lexer.number("36", yytext); }
{ _35 } { Segmenter_p->time_lexer.number("35", yytext); }
{ _34 } { Segmenter_p->time_lexer.number("34", yytext); }
{ _33 } { Segmenter_p->time_lexer.number("33", yytext); }
{ _32 } { Segmenter_p->time_lexer.number("32", yytext); }
{ _31 } { Segmenter_p->time_lexer.number("31", yytext); }
{ _30 } { Segmenter_p->time_lexer.number("30", yytext); }
{ _29 } { Segmenter_p->time_lexer.number("29", yytext); }
{ _28 } { Segmenter_p->time_lexer.number("28", yytext); }
{ _27 } { Segmenter_p->time_lexer.number("27", yytext); }
{ _26 } { Segmenter_p->time_lexer.number("26", yytext); }
{ _25 } { Segmenter_p->time_lexer.number("25", yytext); }
{ _24 } { Segmenter_p->time_lexer.number("24", yytext); }
{ _23 } { Segmenter_p->time_lexer.number("23", yytext); }
{ _22 } { Segmenter_p->time_lexer.number("22", yytext); }
{ _21 } { Segmenter_p->time_lexer.number("21", yytext); }
{ _20 } { Segmenter_p->time_lexer.number("20", yytext); }
{ _19 } { Segmenter_p->time_lexer.number("19", yytext); }
{ _18 } { Segmenter_p->time_lexer.number("18", yytext); }
{ _17 } { Segmenter_p->time_lexer.number("17", yytext); }
{ _16 } { Segmenter_p->time_lexer.number("16", yytext); }
{ _15 } { Segmenter_p->time_lexer.number("15", yytext); }
{ _14 } { Segmenter_p->time_lexer.number("14", yytext); }
{ _13 } { Segmenter_p->time_lexer.number("13", yytext); }
{ _12 } { Segmenter_p->time_lexer.number("12", yytext); }
{ _11 } { Segmenter_p->time_lexer.number("11", yytext); }
{ _10 } { Segmenter_p->time_lexer.number("10", yytext); }
{ _9_1 } { Segmenter_p->time_lexer.number("9.1", yytext); }
{ _9_2 } { Segmenter_p->time_lexer.number("9.2", yytext); }
{ _9_25 } { Segmenter_p->time_lexer.number("9.25", yytext); }
{ _9_25HR } { Segmenter_p->time_lexer.number("9.25", yytext); }
           { Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _9_25DA } { Segmenter_p->time_lexer.number("9.25", yytext); }

```

```

Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _9_25WK } { Segmenter_p->time_lexer.number("9.25", yytext);
Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _9_3 } { Segmenter_p->time_lexer.number("9.3", yytext); }
{ _9_4 } { Segmenter_p->time_lexer.number("9.4", yytext); }
{ _9_5 } { Segmenter_p->time_lexer.number("9.5", yytext); }
{ _9_5HR } { Segmenter_p->time_lexer.number("9.5", yytext);
Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _9_5DA } { Segmenter_p->time_lexer.number("9.5", yytext);
Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _9_5WK } { Segmenter_p->time_lexer.number("9.5", yytext);
Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _9_6 } { Segmenter_p->time_lexer.number("9.6", yytext); }
{ _9_7 } { Segmenter_p->time_lexer.number("9.7", yytext); }
{ _9_75 } { Segmenter_p->time_lexer.number("9.75", yytext); }
{ _9_8 } { Segmenter_p->time_lexer.number("9.8", yytext); }
{ _9_9 } { Segmenter_p->time_lexer.number("9.9", yytext); }
{ _9 } { Segmenter_p->time_lexer.number("9", yytext); }
{ _8_1 } { Segmenter_p->time_lexer.number("8.1", yytext); }
{ _8_2 } { Segmenter_p->time_lexer.number("8.2", yytext); }
{ _8_25 } { Segmenter_p->time_lexer.number("8.25", yytext); }
{ _8_25HR } { Segmenter_p->time_lexer.number("8.25", yytext);
Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _8_25DA } { Segmenter_p->time_lexer.number("8.25", yytext);
Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _8_25WK } { Segmenter_p->time_lexer.number("8.25", yytext);
Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _8_3 } { Segmenter_p->time_lexer.number("8.3", yytext); }
{ _8_4 } { Segmenter_p->time_lexer.number("8.4", yytext); }
{ _8_5 } { Segmenter_p->time_lexer.number("8.5", yytext); }
{ _8_5HR } { Segmenter_p->time_lexer.number("8.5", yytext);
Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _8_5DA } { Segmenter_p->time_lexer.number("8.5", yytext);
Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _8_5WK } { Segmenter_p->time_lexer.number("8.5", yytext);
Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _8_6 } { Segmenter_p->time_lexer.number("8.6", yytext); }
{ _8_7 } { Segmenter_p->time_lexer.number("8.7", yytext); }
{ _8_75 } { Segmenter_p->time_lexer.number("8.75", yytext); }
{ _8_8 } { Segmenter_p->time_lexer.number("8.8", yytext); }
{ _8_9 } { Segmenter_p->time_lexer.number("8.9", yytext); }
{ _8 } { Segmenter_p->time_lexer.number("8", yytext); }
{ _7_1 } { Segmenter_p->time_lexer.number("7.1", yytext); }
{ _7_2 } { Segmenter_p->time_lexer.number("7.2", yytext); }
{ _7_25 } { Segmenter_p->time_lexer.number("7.25", yytext); }
{ _7_25HR } { Segmenter_p->time_lexer.number("7.25", yytext);
Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _7_25DA } { Segmenter_p->time_lexer.number("7.25", yytext);
Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _7_25WK } { Segmenter_p->time_lexer.number("7.25", yytext);
Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _7_3 } { Segmenter_p->time_lexer.number("7.3", yytext); }
{ _7_4 } { Segmenter_p->time_lexer.number("7.4", yytext); }
{ _7_5 } { Segmenter_p->time_lexer.number("7.5", yytext); }
{ _7_5HR } { Segmenter_p->time_lexer.number("7.5", yytext);
Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _7_5DA } { Segmenter_p->time_lexer.number("7.5", yytext);

```

```

Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _7_5WK} { Segmenter_p->time_lexer.number("7.5", yytext);
Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _7_6} { Segmenter_p->time_lexer.number("7.6", yytext); }
{ _7_7} { Segmenter_p->time_lexer.number("7.7", yytext); }
{ _7_75} { Segmenter_p->time_lexer.number("7.75", yytext); }
{ _7_8} { Segmenter_p->time_lexer.number("7.8", yytext); }
{ _7_9} { Segmenter_p->time_lexer.number("7.9", yytext); }
{ _7} { Segmenter_p->time_lexer.number("7", yytext); }
{ _6_1} { Segmenter_p->time_lexer.number("6.1", yytext); }
{ _6_2} { Segmenter_p->time_lexer.number("6.2", yytext); }
{ _6_25} { Segmenter_p->time_lexer.number("6.25", yytext); }
{ _6_25HR} { Segmenter_p->time_lexer.number("6.25", yytext);
Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _6_25DA} { Segmenter_p->time_lexer.number("6.25", yytext);
Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _6_25WK} { Segmenter_p->time_lexer.number("6.25", yytext);
Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _6_3} { Segmenter_p->time_lexer.number("6.3", yytext); }
{ _6_4} { Segmenter_p->time_lexer.number("6.4", yytext); }
{ _6_5} { Segmenter_p->time_lexer.number("6.5", yytext); }
{ _6_5HR} { Segmenter_p->time_lexer.number("6.5", yytext);
Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _6_5DA} { Segmenter_p->time_lexer.number("6.5", yytext);
Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _6_5WK} { Segmenter_p->time_lexer.number("6.5", yytext);
Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _6_6} { Segmenter_p->time_lexer.number("6.6", yytext); }
{ _6_7} { Segmenter_p->time_lexer.number("6.7", yytext); }
{ _6_75} { Segmenter_p->time_lexer.number("6.75", yytext); }
{ _6_8} { Segmenter_p->time_lexer.number("6.8", yytext); }
{ _6_9} { Segmenter_p->time_lexer.number("6.9", yytext); }
{ _6} { Segmenter_p->time_lexer.number("6", yytext); }
{ _5_1} { Segmenter_p->time_lexer.number("5.1", yytext); }
{ _5_2} { Segmenter_p->time_lexer.number("5.2", yytext); }
{ _5_25} { Segmenter_p->time_lexer.number("5.25", yytext); }
{ _5_25HR} { Segmenter_p->time_lexer.number("5.25", yytext);
Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _5_25DA} { Segmenter_p->time_lexer.number("5.25", yytext);
Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _5_25WK} { Segmenter_p->time_lexer.number("5.25", yytext);
Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _5_3} { Segmenter_p->time_lexer.number("5.3", yytext); }
{ _5_4} { Segmenter_p->time_lexer.number("5.4", yytext); }
{ _5_5} { Segmenter_p->time_lexer.number("5.5", yytext); }
{ _5_5HR} { Segmenter_p->time_lexer.number("5.5", yytext);
Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _5_5DA} { Segmenter_p->time_lexer.number("5.5", yytext);
Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _5_5WK} { Segmenter_p->time_lexer.number("5.5", yytext);
Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _5_6} { Segmenter_p->time_lexer.number("5.6", yytext); }
{ _5_7} { Segmenter_p->time_lexer.number("5.7", yytext); }
{ _5_75} { Segmenter_p->time_lexer.number("5.75", yytext); }
{ _5_8} { Segmenter_p->time_lexer.number("5.8", yytext); }
{ _5_9} { Segmenter_p->time_lexer.number("5.9", yytext); }
{ _5} { Segmenter_p->time_lexer.number("5", yytext); }

```

```

{ _4_1 } { Segmenter_p->time_lexer.number("4.1", yytext); }
{ _4_2 } { Segmenter_p->time_lexer.number("4.2", yytext); }
{ _4_25 } { Segmenter_p->time_lexer.number("4.25", yytext); }
{ _4_25HR } { Segmenter_p->time_lexer.number("4.25", yytext); }
{ Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _4_25DA } { Segmenter_p->time_lexer.number("4.25", yytext); }
{ Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _4_25WK } { Segmenter_p->time_lexer.number("4.25", yytext); }
{ Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _4_3 } { Segmenter_p->time_lexer.number("4.3", yytext); }
{ _4_4 } { Segmenter_p->time_lexer.number("4.4", yytext); }
{ _4_5 } { Segmenter_p->time_lexer.number("4.5", yytext); }
{ _4_5HR } { Segmenter_p->time_lexer.number("4.5", yytext); }
{ Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _4_5DA } { Segmenter_p->time_lexer.number("4.5", yytext); }
{ Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _4_5WK } { Segmenter_p->time_lexer.number("4.5", yytext); }
{ Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _4_6 } { Segmenter_p->time_lexer.number("4.6", yytext); }
{ _4_7 } { Segmenter_p->time_lexer.number("4.7", yytext); }
{ _4_75 } { Segmenter_p->time_lexer.number("4.75", yytext); }
{ _4_8 } { Segmenter_p->time_lexer.number("4.8", yytext); }
{ _4_9 } { Segmenter_p->time_lexer.number("4.9", yytext); }
{ _4 } { Segmenter_p->time_lexer.number("4", yytext); }
{ _3_1 } { Segmenter_p->time_lexer.number("3.1", yytext); }
{ _3_2 } { Segmenter_p->time_lexer.number("3.2", yytext); }
{ _3_25 } { Segmenter_p->time_lexer.number("3.25", yytext); }
{ _3_25HR } { Segmenter_p->time_lexer.number("3.25", yytext); }
{ Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _3_25DA } { Segmenter_p->time_lexer.number("3.25", yytext); }
{ Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _3_25WK } { Segmenter_p->time_lexer.number("3.25", yytext); }
{ Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _3_3 } { Segmenter_p->time_lexer.number("3.3", yytext); }
{ _3_4 } { Segmenter_p->time_lexer.number("3.4", yytext); }
{ _3_5 } { Segmenter_p->time_lexer.number("3.5", yytext); }
{ _9_5HR } { Segmenter_p->time_lexer.number("3.5", yytext); }
{ Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _9_5DA } { Segmenter_p->time_lexer.number("3.5", yytext); }
{ Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _9_5WK } { Segmenter_p->time_lexer.number("3.5", yytext); }
{ Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _3_6 } { Segmenter_p->time_lexer.number("3.6", yytext); }
{ _3_7 } { Segmenter_p->time_lexer.number("3.7", yytext); }
{ _3_75 } { Segmenter_p->time_lexer.number("3.75", yytext); }
{ _3_8 } { Segmenter_p->time_lexer.number("3.8", yytext); }
{ _3_9 } { Segmenter_p->time_lexer.number("3.9", yytext); }
{ _3 } { Segmenter_p->time_lexer.number("3", yytext); }
{ _2_1 } { Segmenter_p->time_lexer.number("2.1", yytext); }
{ _2_2 } { Segmenter_p->time_lexer.number("2.2", yytext); }
{ _2_25 } { Segmenter_p->time_lexer.number("2.25", yytext); }
{ _2_25HR } { Segmenter_p->time_lexer.number("2.25", yytext); }
{ Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _2_25DA } { Segmenter_p->time_lexer.number("2.25", yytext); }
{ Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _2_25WK } { Segmenter_p->time_lexer.number("2.25", yytext); }
{ Segmenter_p->time_lexer.temporal("WK", yytext); }

```

```

{ _2_3 } { Segmenter_p->time_lexer.number("2.3", yytext); }
{ _2_4 } { Segmenter_p->time_lexer.number("2.4", yytext); }
{ _2_5 } { Segmenter_p->time_lexer.number("2.5", yytext); }
{ _2_5HR } { Segmenter_p->time_lexer.number("2.5", yytext); }
           { Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _2_5DA } { Segmenter_p->time_lexer.number("2.5", yytext); }
           { Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _2_5WK } { Segmenter_p->time_lexer.number("2.5", yytext); }
           { Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _2_6 } { Segmenter_p->time_lexer.number("2.6", yytext); }
{ _2_7 } { Segmenter_p->time_lexer.number("2.7", yytext); }
{ _2_75 } { Segmenter_p->time_lexer.number("2.75", yytext); }
{ _2_8 } { Segmenter_p->time_lexer.number("2.8", yytext); }
{ _2_9 } { Segmenter_p->time_lexer.number("2.9", yytext); }
{ _2 } { Segmenter_p->time_lexer.number("2", yytext); }
{ _1_1 } { Segmenter_p->time_lexer.number("1.1", yytext); }
{ _1_2 } { Segmenter_p->time_lexer.number("1.2", yytext); }
{ _1_25 } { Segmenter_p->time_lexer.number("1.25", yytext); }
{ _1_25HR } { Segmenter_p->time_lexer.number("1.25", yytext); }
           { Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _1_25DA } { Segmenter_p->time_lexer.number("1.25", yytext); }
           { Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _1_25WK } { Segmenter_p->time_lexer.number("1.25", yytext); }
           { Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _1_3 } { Segmenter_p->time_lexer.number("1.3", yytext); }
{ _1_4 } { Segmenter_p->time_lexer.number("1.4", yytext); }
{ _1_5 } { Segmenter_p->time_lexer.number("1.5", yytext); }
{ _1_5HR } { Segmenter_p->time_lexer.number("1.5", yytext); }
           { Segmenter_p->time_lexer.temporal("HR", yytext); }
{ _1_5DA } { Segmenter_p->time_lexer.number("1.5", yytext); }
           { Segmenter_p->time_lexer.temporal("DA", yytext); }
{ _1_5WK } { Segmenter_p->time_lexer.number("1.5", yytext); }
           { Segmenter_p->time_lexer.temporal("WK", yytext); }
{ _1_6 } { Segmenter_p->time_lexer.number("1.6", yytext); }
{ _1_7 } { Segmenter_p->time_lexer.number("1.7", yytext); }
{ _1_75 } { Segmenter_p->time_lexer.number("1.75", yytext); }
{ _1_8 } { Segmenter_p->time_lexer.number("1.8", yytext); }
{ _1_9 } { Segmenter_p->time_lexer.number("1.9", yytext); }
{ _1 } { Segmenter_p->time_lexer.number("1", yytext); }
{ _0_1 } { Segmenter_p->time_lexer.number("0.1", yytext); }
{ _0_2 } { Segmenter_p->time_lexer.number("0.2", yytext); }
{ _0_25 } { Segmenter_p->time_lexer.number("0.25", yytext); }
{ _0_3 } { Segmenter_p->time_lexer.number("0.3", yytext); }
{ _0_4 } { Segmenter_p->time_lexer.number("0.4", yytext); }
{ _0_5 } { Segmenter_p->time_lexer.number("0.5", yytext); }
{ _0_6 } { Segmenter_p->time_lexer.number("0.6", yytext); }
{ _0_7 } { Segmenter_p->time_lexer.number("0.7", yytext); }
{ _0_75 } { Segmenter_p->time_lexer.number("0.75", yytext); }
{ _0_8 } { Segmenter_p->time_lexer.number("0.8", yytext); }
{ _0_9 } { Segmenter_p->time_lexer.number("0.9", yytext); }

[^\.\n] { Segmenter_p->time_lexer.not_temporal(yytext); }
\. { Segmenter_p->time_lexer.not_temporal(yytext); }
   { Segmenter_p->time_lexer.clear(); }
\n { Segmenter_p->time_lexer.end_line(); }

```

%%


```
int yyposition() { return (strlen((char *)YY_CURRENT_BUFFER-
>yy_buf_pos)); }

#ifdef LEXER
SegmenterBaseClass Segmenter;
SegmenterBaseClass *Segmenter_p;

int
main(int argc, char* argv[]) {
    Segmenter_p = &Segmenter;
    return yylex();
}
#endif
```